

Dokumentace k projektu z IPP, zadání DKA Determinizace konečného automatu

1. Analýza

V projektu bylo za úkol napsat skript v jazyce PHP, který načte obecný konečný automat a převede jej na ekvivalentní deterministický automat.

2. Řešení

2.1. Parametry

Funkce zpracovává parametry příkazové řádky a porovnává jednotlivé parametry pomocí funkcí `strcmp` a `substr`. Funkce zároveň hlídá všechny kolidující parametry a případně volá funkci `errors()` s kódem jedna pro ukončení skriptu s chybou.

2.2. Načítání vstupu

Pomocí funkce `inputFile()` načítám vstup znak po znaku a jednotlivé hodnoty ukládám do polí pro jednotlivé prvky konečného automatu. Funkce taky vykonává lexikální a syntaktickou analýzu vstupního konečného automatu přičemž vynechává bílé znaky a taky komentáře.

2.3. Epsilon přechody

Pro správnost determinizace je potřeba nejdřív odstranit z automatu všechny epsilon přechody. Skript tedy pomocí rekurzivního volání funkce vytvoří epsilon uzávěry, které pak pomocí algoritmu z předmětu IFJ přetvoří na nové přechody a tedy na nedeterministický konečný automat bez epsilon přechodů. Funkce taky vytváří nové koncové stavy.

2.4. Determinizace

Po odstranění epsilon přechodů pak může dojít k samotné determinizaci a to opět pomocí rekurzivního volání funkce obsahující algoritmus z předmětu IFJ na vytvoření deterministického konečného automatu bez generování nedostupných stavů. Využívá se tedy taky spájení stavů, a tedy vznikají nové stavy složené ze stavů, které spájíme oddělené znakem "_". Zároveň vznikají nové koncové stavy za předpokladu, že byli vytvořené spojením s alespoň jedním z původních stavů.

3. Rozšíření

3.1. WCH

Upravuje jednotlivé části načítací funkce pro úpravu syntaktické analýzy a zápis jednotlivých parametrů vstupního konečného automatu. Na vstupu je tedy možné brát bílé znaky či komentáře jako oddělovací čárku a taky je možné vynechávat apostrofy.

3.2. RUL

Vstup obsahuje pouze pravidla daného konečného automatu, z kterých si pak funkce `rul_only()` vytvoří ostatní parametry konečného automatu.

3.3. STR

Zadaný vstupní řetězec je zkontrolován, zda je řetězcem jazyka vstupního konečného automatu.

V případě že některý ze znaků řetězce se nenachází v abecedě vstupního konečného automatu, dochází k chybě s kódem 1. V opačném případě dochází k správné funkčnosti skriptu s návratovým kódem 0, přičemž za předpokladu, že daný řetězec je řetězcem jazyka vstupního automatu, bude na výstup vypsána 1. Pokud ne, bude vypsána 0.

3.4. Testování

Na testování sem využil v první řadě skript poskytnut k zadání projektu a pak vlastní skript v BASH s přibližně třiceti testy na různé chyby či správné vstupy konečného automatu.